

Олимпиада «Ломоносов» по информатике
2025-2026 учебный год. Заключительный этап
Работа участника с id заявки 1619727, логином inf26f_200

Сводный итог по всем задачам в проверяющей системе

Run ID	Time	User name	Problem	Language	Result	Tests	Score
1311	3:58:11	inf26f_200	6	g++	Partial solution	0	0
718	2:39:46	inf26f_200	5	g++	OK	103	100
640	2:27:13	inf26f_200	4	g++	OK	103	100
607	2:21:38	inf26f_200	3	g++	Partial solution	20	85
584	2:18:48	inf26f_200	2	g++	OK	53	100
570	2:16:13	inf26f_200	1	pyru3	OK	23	100

485 (четыреста восемьдесят пять) технических баллов
69 (шестьдесят девять) итоговых баллов



председатель Жюри кандидат физ.-мат. наук Малышко В. В.



зам. председателя Жюри кандидат физ.-мат. наук Корныхин Е. В.

Посылка по задаче 1

```
[1] k = int(input())
[2] mp = {'N': 0, 'I': 1, 'X': 10, 'V': 5, 'L': 50}
[3] mn = float('inf')
[4] mx = float('-inf')
[5] for i in range(k):
[6]     s = input()
[7]     next = '.'
[8]     num = []
[9]     curr = 0
[10]    for j in range(len(s) - 1, -1, -1):
[11]        if s[j] == '.':
[12]            num.append(curr)
[13]            curr = 0
[14]            next = '.'
[15]        elif next == '.' or mp[next] <= mp[s[j]]:
[16]            curr += mp[s[j]]
[17]            next = s[j]
[18]        else:
[19]            curr -= mp[s[j]]
[20]
[21]    num.append(curr)
[22]    num.reverse()
[23]    t = 0
[24]    for d in num:
[25]        t = t * 52 + d
[26]
[27]    mn = min(mn, t)
[28]    mx = max(mx, t)
[29]
[30] amin = []
[31] amax = []
[32] while mn > 0:
[33]     if mn % 52 < 26:
[34]         amin.append(chr(ord('A') + mn % 52))
[35]     else:
[36]         amin.append(chr(ord('a') + mn % 52 - 26))
[37]     mn //= 52
[38]
[39] while mx > 0:
[40]     if mx % 52 < 26:
[41]         amax.append(chr(ord('A') + mx % 52))
[42]     else:
[43]         amax.append(chr(ord('a') + mx % 52 - 26))
[44]     mx //= 52
[45]
[46] amin.reverse()
[47] amax.reverse()
[48]
[49] res1 = ''.join(amin)
[50] res2 = ''.join(amax)
[51]
[52] if res1 == '':
[53]     res1 = 'A'
[54] if res2 == '':
[55]     res2 = 'A'
[56]
[57] print(res1)
[58] print(res2)
[59]
```

Посылка по задаче 2

```
[1] #include <iostream>
[2] #include <vector>
[3] #include <random>
[4] #include <queue>
[5] #include <numeric>
[6] #include <cmath>
[7] #include <utility>
[8] #include <map>
[9] #include <set>
[10] #include <unordered_map>
[11] #include <unordered_set>
[12] #include <array>
[13] #include <algorithm>
[14]
[15] using ll = long long;
[16] using ld = long double;
[17] const ll INF = 1e18, NINF = -1e18;
[18] const ld EPS = 1e-8;
[19]
[20] using namespace std;
[21]
[22] int main()
[23] {
[24]     ios_base::sync_with_stdio(false);
[25]     cin.tie(nullptr);
[26]
[27]     ll n;
[28]     cin >> n;
[29]
[30]     vector<ll> counter(160001);
[31]     vector<bool> is_brazil(160001), is_prime(160001, true);
[32]     is_prime[0] = is_prime[1] = false;
[33]     for (ll i = 2; i <= 160000; i++) {
[34]         if (!is_prime[i])
[35]             continue;
[36]
[37]         for (ll j = i * i; j <= 160000; j += i) {
[38]             is_prime[j] = false;
[39]         }
[40]     }
[41]
[42]     for (ll b = 2; b <= 160000; b++) {
[43]         ll curr = (1 * b + 1) * b + 1;
[44]         while (curr <= 160000) {
[45]             is_brazil[curr] = true;
[46]             curr = curr * b + 1;
[47]         }
[48]     }
[49]     for (ll i = 0; i < n; i++) {
[50]         ll num;
[51]         cin >> num;
[52]         if (is_brazil[num] && is_prime[num])
[53]             ++counter[num];
[54]     }
[55]     ll max_num = 0, max_count = 0;
[56]     for (ll i = 160000; i >= 0; i--) {
[57]         if (counter[i] > max_count) {
[58]             max_num = i;
[59]             max_count = counter[i];
[60]         }
[61]     }
[62]
[63]     cout << max_num;
[64]
[65]     return 0;
[66] }
```

Посылка по задаче 3

```
[1] #include <bits/stdc++.h>
[2]
[3] using namespace std;
[4]
[5] using ll = long long;
[6]
[7] int main() {
[8]     ios_base::sync_with_stdio(false);
[9]     cin.tie(nullptr);
[10]
[11]     ll n, i, j;
[12]     cin >> n >> j >> i;
[13]     ll offset = 0;
[14]     --i;
[15]     --j;
[16]
[17]     ll passed = 0;
[18]     while (n > 0) {
[19]         if (j == 0) {
[20]             passed += i;
[21]             break;
[22]         }
[23]         passed += n - 1;
[24]         if (i == n - 1) {
[25]             passed += j;
[26]             break;
[27]         }
[28]         passed += n - 1;
[29]         if (j == n - 1) {
[30]             passed += n - 1 + i;
[31]             break;
[32]         }
[33]         passed += n - 1;
[34]         if (i == 0) {
[35]             passed += n - 1 - j;
[36]             break;
[37]         }
[38]         passed += n - 1;
[39]         --i;
[40]         --j;
[41]         n -= 2;
[42]     }
[43]
[44]     cout << passed;
[45]     return 0;
[46] }
```

Посылка по задаче 4

```
[1] #include <bits/stdc++.h>
[2]
[3] using namespace std;
[4]
[5] using ll = long long;
[6]
[7] int main() {
[8]     ios_base::sync_with_stdio(false);
[9]     cin.tie(nullptr);
[10]
[11]     ll n;
[12]     cin >> n;
[13]     vector<ll> arr(n);
[14]
[15]     for (int i = 0; i < n; i++) {
[16]         cin >> arr[i];
[17]     }
[18]
[19]     reverse(arr.begin(), arr.end());
[20]
[21]     vector<ll> curr;
[22]     for (int i = 0; i < n; i++) {
[23]         ll ind = upper_bound(curr.begin(), curr.end(), arr[i]) - curr.begin();
[24]         if (ind >= curr.size()) {
[25]             curr.emplace_back(arr[i]);
[26]         } else {
[27]             curr[ind] = arr[i];
[28]         }
[29]     }
[30]
[31]     ll res = arr.size() - curr.size();
[32]
[33]     cout << res;
[34]
[35]
[36]     return 0;
[37] }
```

Посылка по задаче 5

```
[1] #include <bits/stdc++.h>
[2]
[3] using namespace std;
[4]
[5] using ll = long long;
[6]
[7] vector<bool> used;
[8] vector<vector<ll>> adj;
[9] vector<ll> topsort;
[10]
[11] void dfs(ll curr) {
[12]     if (used[curr]) {
[13]         return;
[14]     }
[15]
[16]     used[curr] = true;
[17]     for (ll next : adj[curr]) {
[18]         dfs(next);
[19]     }
[20]
[21]     topsort.emplace_back(curr);
[22] }
[23]
[24] int main() {
[25]     ios_base::sync_with_stdio(false);
[26]     cin.tie(nullptr);
[27]
[28]
[29]     ll n;
[30]     cin >> n;
[31]
[32]     vector<vector<ll>> arr(n, vector<ll>(3));
[33]     for (ll i = 0; i < n; i++) {
[34]         cin >> arr[i][0] >> arr[i][1] >> arr[i][2];
[35]     }
[36]
[37]     adj.resize(n);
[38]     used.resize(n);
[39]
[40]     for (ll i = 0; i < n; i++) {
[41]         ll x1 = arr[i][0], y1 = arr[i][1], h1 = arr[i][2];
[42]         for (ll j = 0; j < n; j++) {
[43]             ll x2 = arr[j][0], y2 = arr[j][1], h2 = arr[j][2];
[44]             if (x1 < x2 && y1 < y2 && h1 > h2) {
[45]                 adj[i].emplace_back(j);
[46]             }
[47]         }
[48]     }
[49]
[50]     topsort.reserve(n);
[51]     for (ll i = 0; i < n; i++)
[52]         dfs(i);
[53]
[54]     vector<ll> dp(n);
[55]     ll res = 0;
[56]     for (ll ind : topsort) {
[57]         for (ll next : adj[ind]) {
[58]             dp[ind] = max(dp[ind], dp[next] + 1);
[59]         }
[60]         res = max(res, dp[ind]);
[61]     }
[62]
[63]     cout << res + 1;
[64]
[65]     return 0;
[66] }
```

Посылка по задаче 6

```
[1] #include <bits/stdc++.h>
[2]
[3] using namespace std;
[4]
[5] using ll = long long;
[6]
[7] //pragma GCC optimize("avx2, unroll-loops")
[8]
[9] vector<bool> board;
[10] map<pair<vector<bool>, ll>, ll> dp;
[11] vector<ll> ships;
[12] ll r, c;
[13] ll countt = 0;
[14]
[15] ll search() {
[16]     if (countt == 0) {
[17]         return 1;
[18]     }
[19]
[20]     ll hash = 0;
[21]     for (int i = 1; i < 5; i++) {
[22]         hash = hash * 5 + ships[i];
[23]     }
[24]
[25]     pair<vector<bool>, ll> key = {board, hash};
[26]     if (dp.count(key)) {
[27]         return dp[key];
[28]     }
[29]
[30]     if (hash == 126) {
[31]         int ii = 0;
[32]     }
[33]
[34]     ll res = 0;
[35]     ll its = 0;
[36]     for (int sz = 1; sz < 5; sz++) {
[37]         if (ships[sz] == 0)
[38]             continue;
[39]         for (int i = 0; i < r; i++) {
[40]             ll curr_len = 0;
[41]             for (int j = 0; j < c; j++) {
[42]                 ll ind = i * c + j;
[43]                 if (!board[ind]) {
[44]                     for (int k = j - curr_len; k < j; k++) {
[45]                         board[i * c + k] = true;
[46]                         ++countt;
[47]                     }
[48]                     curr_len = 0;
[49]                     continue;
[50]                 }
[51]
[52]                 curr_len++;
[53]                 --countt;
[54]                 board[ind] = false;
[55]                 if (curr_len == sz) {
[56]                     --ships[sz];
[57]                     res += search();
[58]                     ++ships[sz];
[59]                     break;
[60]                 }
[61]             }
[62]
[63]             for (int k = c - curr_len; k < c; k++) {
[64]                 board[i * c + k] = true;
[65]                 ++countt;
[66]             }
[67]         }
[68]     }
[69]
[70]     for (int sz = 2; sz < 5; sz++) {
[71]         if (ships[sz] == 0)
[72]             continue;
[73]         for (int j = 0; j < c; j++) {
[74]             ll curr_len = 0;
[75]             for (int i = 0; i < r; i++) {
```

```

[76]         ll ind = i * c + j;
[77]         if (!board[ind]) {
[78]             for (int k = i - curr_len; k < i; k++) {
[79]                 board[k * c + j] = true;
[80]                 ++countt;
[81]             }
[82]             curr_len = 0;
[83]             continue;
[84]         }
[85]
[86]         curr_len++;
[87]         --countt;
[88]         board[ind] = false;
[89]         if (curr_len == sz) {
[90]             --ships[sz];
[91]             res += search();
[92]             ++ships[sz];
[93]             break;
[94]         }
[95]     }
[96]
[97]     for (int k = r - curr_len; k < r; k++) {
[98]         board[k * c + j] = true;
[99]         ++countt;
[100]    }
[101] }
[102] }
[103]
[104] dp[key] = res;
[105]
[106] return res;
[107] }
[108]
[109]
[110] int main() {
[111]     ios_base::sync_with_stdio(false);
[112]     cin.tie(nullptr);
[113]
[114]     ships.resize(5);
[115]     for (int i = 1; i <= 4; i++) {
[116]         cin >> ships[i];
[117]     }
[118]     cin >> r >> c;
[119]
[120]     board.resize(r * c);
[121]     for (int i = 0; i < r; i++) {
[122]         for (int j = 0; j < c; j++) {
[123]             char ch;
[124]             cin >> ch;
[125]             board[i * c + j] = (ch == '#');
[126]             countt += (ch == '#');
[127]         }
[128]     }
[129]
[130]     ll res = search();
[131]
[132]     cout << res;
[133]
[134]
[135]     return 0;
[136] }

```

Посылка по задаче 7

Посылок по задаче 7 участником не было отправлено.